

EFFICIENT REALIZATION OF JPEG ENCODER FOR IMAGE COMPRESSION ON FPGA

A. KAVITHA & P. A. HARSHA VARDHINI

Department of ECE, Vignan Institute of Technology & Science, Deshmukhi, Hyderabad, Andhra Pradesh, India

ABSTRACT

The JPEG encoder is a major component in JPEG standard which is used in image compression. A complex sub-block DCT (Discrete Cosine Transform) is involved with other coding blocks such as Zigzag, Entropy and Quantization. Using two 1 dimensional DCTs connected by a transpose buffer, 2 dimensional DCT is computed. Hardware implementation of pipelined 2 dimensional DCT is designed in this work. The architecture uses 4059 slices, 6885 LUT, 58 I/Os of Xilinx Spartan-6 XC6SLX16 FPGA and works at an operating frequency of 120 MHz.

KEYWORDS: DCT, JPEG Encoder, Zigzag, Quantization, VLC, Compression Ratio

INTRODUCTION

In this real world application, Image Compression plays an important role in the digital world. Applications like commercial photography, industrial imagery or video comes under this category of applications. For these various applications, various needs are to be met, the JPEG standard which includes two basic compression methods where each method has a different mode of operation. A DCT based method is specified for lossy compression, and a predictive method for lossless compression.

For different electronic applications, DCT as a mathematical tool is used. For an efficient application realization like memory savings, fast transmission and compact representation DCT can be used where it transforms the information from time domain and space domain to frequency domain. Five major steps are followed in a JPEG compression procedure. [3]

- Color space conversion
- Down sampling
- 2 dimensional DCT
- Quantization
- Entropy coding

Figure 1 show the last three steps involved in Gray scale image compression. In this paper a pipe lined architecture for 2D-DCT along with quantization, zigzag, and entropy coding modules is implemented. The DCT is most critical module in JPEG encoder since it involves the high computation complexity. This paper proposes the efficient FPGA implementation of JPEG encoder with less number of hardware resources. The results from RTL level Verilog design can be reused for an ASIC implementation in the future.

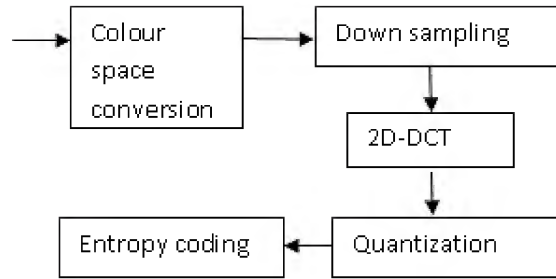


Figure 1: JPEG Compression Steps

IMPLEMENTATION OF IMAGE ENCRYPTION SYSTEM

From many decades, Cryptography is used for secured communication which converts the readable information into coded or unreadable form. This improves the security of multimedia information from unauthorized access. Compared to normal text data, image consists of larger data with higher redundancy and stronger correlation between pixels. As an image posses larger data than a text file, encryption algorithms like DES, RSA used for text encryption algorithms cannot be used for image encryption. But the image encryption got a feasibility that the decrypted image is acceptable because of human perception but decrypted text must match the original text. Many image encryption algorithms are available where some algorithms are prone to attacks. But modern cryptographic mechanism have overcome those limitations.

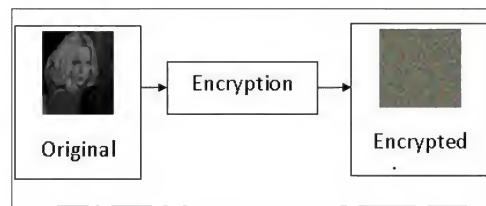


Figure 2: Image Encryption System

Discrete Cosine Transform

Among different transforms DCT got optimum performance with greater energy compaction efficiency. DCT got decorrelation property which gives an advantage of removing redundancy between neighboring pixels and leads to uncorrelated transform coefficients and they can be independently encoded. The two dimensional DCT equation is given by equation (1)

$$C(U, V) = \alpha(U) \alpha(V) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(X, Y) \cos \left[\frac{\pi(2X+1)U}{2N} \right] \cos \left[\frac{\pi(2Y+1)V}{2N} \right] \quad (1)$$

For u, v= 0, 1, 2, ..., N-1.

The inverse transform is defined by Equation (2)

$$f(X, Y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(U) \alpha(V) 2 C(U, V) \cos \left[\frac{\pi(2X+1)U}{2N} \right] \cos \left[\frac{\pi(2Y+1)V}{2N} \right] \quad (2)$$

For x, y= 0, 1, 2, ..., N-1.

The 2 dimensional basis functions are obtained multiplying vertically oriented one dimensional functions with the

horizontal set of same function. The data of the image is split into 8 X 8 blocks of pixels and the processing of each color component is done independently. Though it's a color image, Pixel means the same i.e., 'single value'. With an operating frequency of 166MHz, two dimensional DCT is suitable as core of JPEG compression hardware. Two dimensional DCT is divided using separability property into two one-dimensional DCT calculations by using a transpose buffer.

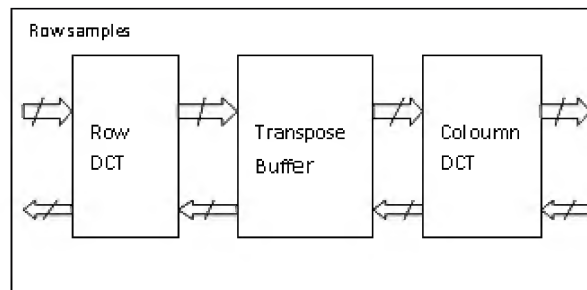


Figure 3: Architecture of 2-D DCT

Figure 3 Shows the Architecture of 2-d Dct. 2d-dct/idct Design Is Divided Into Three Major Blocks Namely Row- Dct, Transpose Buffer, and Column-dct. Row-dct and Column-dct Contains Both 1ddct (Figure 4) By Row.

During Forward transform, 1D-DCT structure (Figure 4) is functionally active. In every cycle two eight bit samples are received as input by the row DCT block and hence its value ranges from -128 to 127. The transformed sample bit width is maintained as 10 bit to accommodate 2 bit increment during 1D-DCT computation architecture (Figure 5) has a four stage internal pipeline shown in Figure 4. Transpose Buffer receives two 10-bit samples as an input every cycle. Each sample is a signed 10-bit value and hence its value ranges from -512 to 511. Since there is no data Manipulation in the module the output sample width remains as input sample width i.e. 10-bit.

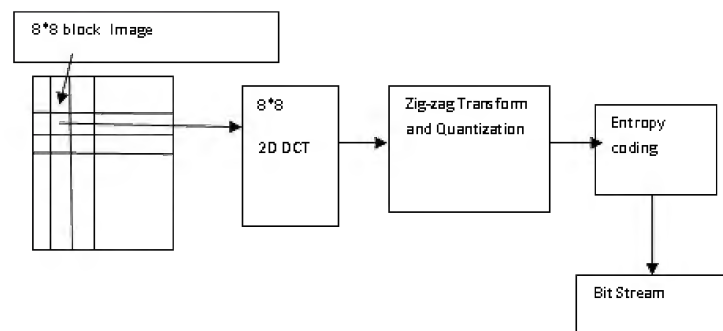


Figure 4: Architecture of 2-D DCT

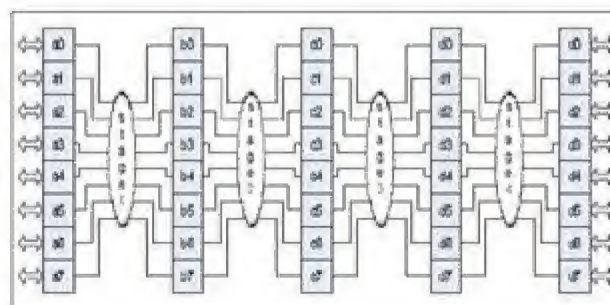


Figure 5: Four Stage Pipeline 1D-DCT

OUTLINE OF JPEG STANDARD

The basic model for the JPEG encoder is shown in Figure 6.

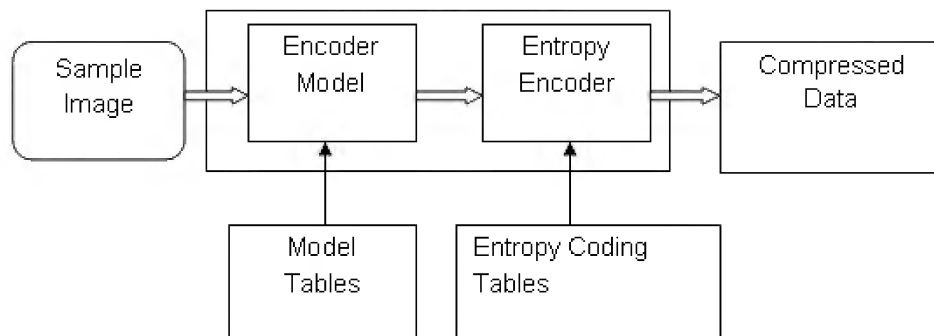


Figure 6: The JPEG Baseline Encoder

The encoder model for the baseline process is shown in Figure 7. The input image is divided into non-overlapping blocks of 8 x 8 pixels, and input to the baseline encoder. The pixel values are converted from unsigned integer format to signed integer format, and DCT computation is performed on each block. DCT transforms the pixel data into a block of spatial frequencies that are called the DCT coefficients. Typically pixels in 8 X 8 neighborhood have small variations in gray levels, DCT output will result in most of the block energy stored in lower spatial frequencies and the higher frequencies does not effect the image quality as it will have values equal to or close to zero and hence can be ignored. JPEG allows for this selection of frequencies.

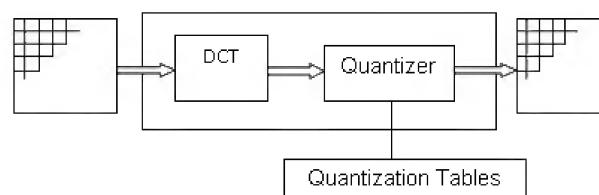


Figure 7: The JPEG Encoder Model

The block of DCT coefficients output by the encoder model is rearranged into one dimensional data by using zigzag reordering as shown in Figure 8.

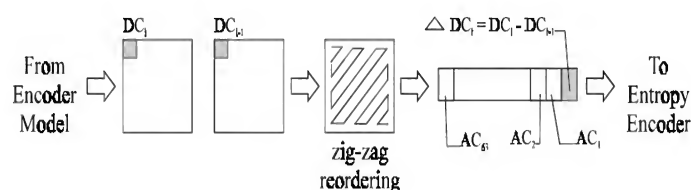


Figure 8: Zigzag Reordering of DCT Output

The location (0, 0) of each block 'i' contains the DC coefficient for the block represented as DC_i . This DC coefficient is replaced by the value ΔDC_i which is the difference between the DC coefficients of block. i. and block 'i. 1'. Since the pixels of adjacent blocks are likely to have similar average energy levels, only the difference between the current and the previous DC coefficients is used, which is commonly known as Differential Pulse Code Modulation (DPCM) technique. It should be noted that the high frequency coefficients that are more likely to be zeroes get grouped at the end of the one-dimensional data due to the zigzag reordering.

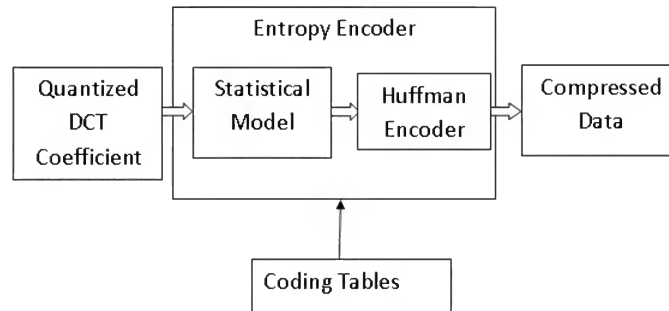


Figure 9: The JPEG Baseline Entropy Encoder

Figure 9 shows the details of entropy encoder and to arrange the DCT coefficients it uses a variable length encoding which depends on a statistical model. In the entropy encoder, the quantized DCT coefficients are converted into a stream of [run length count, category] pairs. For each pair, there is a corresponding variable length Huffman code which will be used by the Huffman encoder to perform the compression. Although the JPEG algorithm is unaffected by color, since it processes each color independently, it has been shown that by changing the color space, the compression ratio can be significantly improved. This is due to the perception of the human visual system and the less perfect characteristics of the display devices. One of the most appropriate color spaces for the JPEG algorithm has been shown to be YC_bC_r , where Y is the luminance component and C_b and C_r are the two chrominance components. Since the luminance component carries much more information compared to the chrominance components, JPEG allows different tables to be used during compression.

JPEG System Architecture

The system architecture for our implementation is shown in Figure 10. The entire architecture is organized as a linear multistage pipeline in order to achieve high throughput. This figure reflects the sequence of computation in the JPEG Baseline process. The architecture consists of the encoder model, and the entropy encoder. The encoder model consists of a DCT module, a quantization module, and reordering logic. The entropy encoder consists of several modules such as the zero run length encoder, category selection unit, Huffman encoder and data packer. The image to be compressed is input to the architecture at the rate of one pixel per clock cycle. The input data is processed by the various modules in a linear fashion, where each module itself is organized internally as a multistage linear pipe. The compressed data is output by the system at a variable rate depending on the amount of compression achieved.

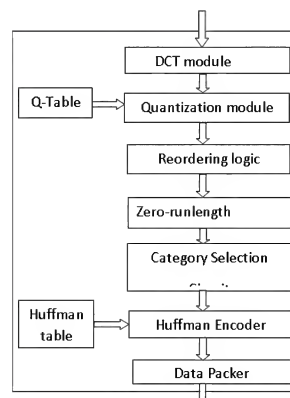


Figure 10: JPEG System Architecture

FPGA IMPLEMENTATION

The system architecture for proposed design is shown in figure 11

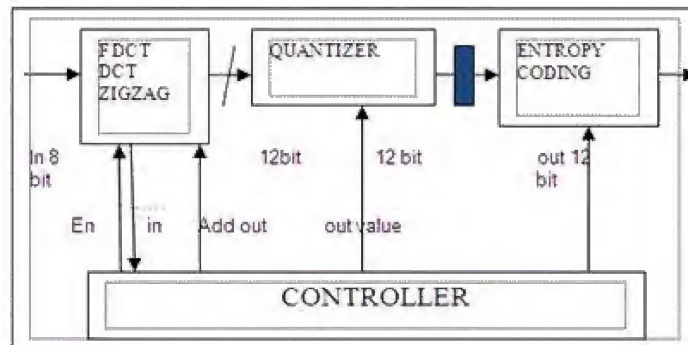


Figure 11: System Architecture

The 2D-DCT architecture is designed based on the design described in [2] and [4]. The quantization and zigzag buffers designed based on the designs described in [3]. The VLC architecture design is based on application note taken from Xilinx [5].

Dimensional DCT Architecture

Vector processing using parallel multipliers is a method used for implementation of DCT. The advantages in the vector processing method are regular structure, simple control and interconnect, and good balance between performance and complexity of implementation. For the case of 8 x 8 block region, a one-dimensional 8- point DCT followed by an internal transpose memory, followed by another one dimensional 8-point DCT provides the 2D DCT architecture.

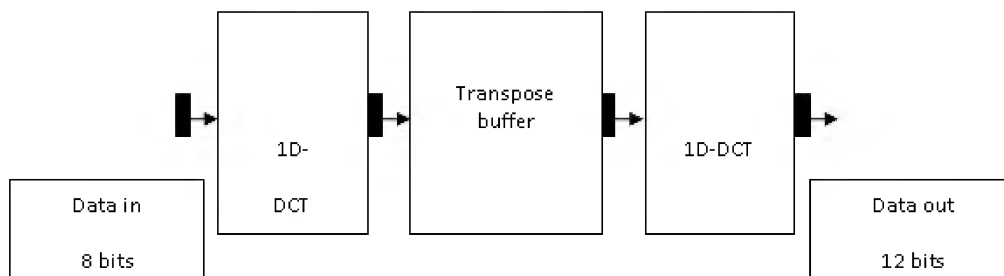


Figure 12: 2-D DCT Architecture

Architecture Construction of Zigzag and Quantizer

The architectures for zigzag and quantizer explained in [2] are used in this paper. The construction of zigzag buffer is like transpose buffer. It has two sets of data - address bus. The input DCT values will be stored in normal sequence like increasing order of address, but the stored values will readout in zigzag sequence. A preprogrammed ROM is used to store the address locations in zigzag sequence as to implement the quantization process; the output of 2DDCT is divided by quantization value. The value is generated by quantization ROM that made for store the quantization table elements. The difference between present work and work done in [2] is it uses division operation instead of multiplication and post scaled values. It uses real time quantization table elements.

VLC Architecture

Variable Length Coding (VLC) is the final lossless stage of the JPEG compression unit. VLC is done to further compress the quantized image. In zigzag scanning the coefficients are read out in a zigzag order. By arranging the coefficients in this manner, RLE and Huffman coding can be done to further compress the data. The scan puts the high frequency components together. These components are usually zeroes. RLE in JPEG consists of denoting the "run of zeroes" followed by the value of the coefficient. Each AC coefficient is represented by the number of zeroes before the coefficient followed by the coefficient. RLE is very simple to implement. A flow chart of an RLE implementation is shown in figure 13.

Implementation Results

The JPEG encoder architecture was described in Verilog. The design was synthesized into a Xilinx Spartan 6 family FPGA. System is tested with real photographic image. The standard grey-scale Lena image shown in figure 14 is used as test bench to verify the design. Data from the picture was converted to Verilog test bench using Matlab technical computation language.

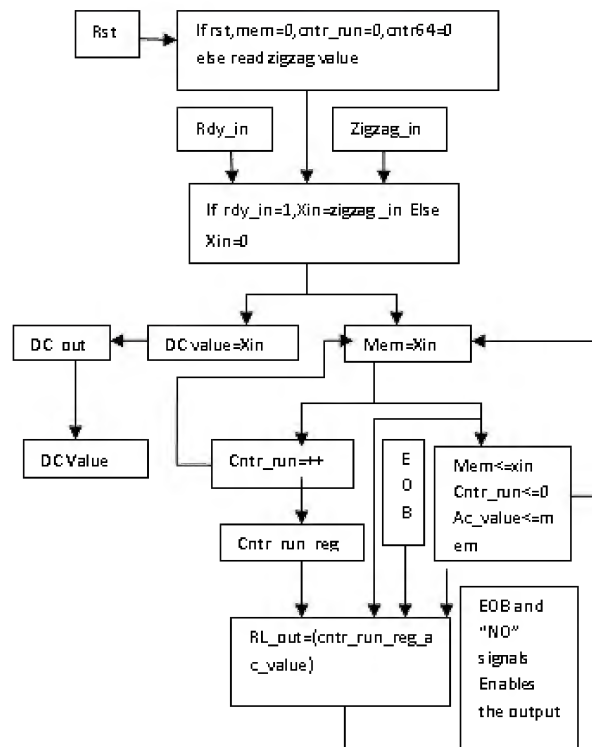


Figure 13: Implementation of RLE



Figure 14: Lena

The complete synthesis results to Spartan-6 FPGA are presented in Table 1, whose hardware was fit in an XC6SLX16 device. The Table 2 presents the comparison between this work 2D-DCT) and the pure 2D-DCT designed in [4]. System designed in [4] uses Virtex FPGA and straightforward multiplication without special algorithm.

Table 1: Device Utilization Using Xilinx Spartan-6

Logic Unit	Used	Available	Utilization
Number of Slices	4, 059	13, 312	30%
Number of slice FFs	4, 018	26, 624	25%
Number of 4 input LUTs	7, 094	26, 624	26%
Number of Bonded IOBs	58	221	26%
Number of BUFGMUXs	1	8	12%

Table 2: Device Utilization Comparison between this Work and 2D-DCT Designed in [4]

Logic Unit	This Work	Presented in [6]
Number of slices	3168	7260
Number of slice FFs	2576	9644
Number of 4 input LUTs	5610	11194
Number of Bonded IOBs	58	101

According to synthesis result, constraint yields minimum clock period 15.025 ns. Maximum clock frequency can be used is 120 MHz. In present paper a complete JPEG encoder module implemented on Spartan6 XC6SLX16 FPGA. The work described in [2] describes the implementation of DCT, quantization and zigzag modules, the work described in [3] only focuses on implementation of 2D-DCT, but in present work another VLC module is included in this work. In work [3] each 8*8 block processed in 5.6 μ s, but in present work each 8*8 block processed in 1.47 μ s. One grey level image completely processed in 1.3ms.

The latency produced by the system in 2D-DCT is 84 clock cycles. Overall system has the latency of 93 clock cycles. As comparison, 2D-DCT designed in [2] has latency 94 clock cycles and 124 clock cycles for overall system. The result reached by system in [3] has 160 clock cycles as system latency to compute 2D-DCT.

CONCLUSIONS

In this work implementation of JPEG encoder architecture for JPEG image compression standard is described. The architectures for the various stages are based on efficient and high performance designs suited for VLSI implementation. The implementation was tested for functional correctness using Verilog with Xilinx tool. The design is tested with grey scale image. Pipeline process causes latency in the system. Maximum frequency can be achieved by this system is 120MHz. The design takes less device resources and suitable for FPGA like Xilinx xc3s1500. The latency produced by design is less compared to previous works. Finally it is designed as a balanced architecture compared to previous works.

REFERENCES

1. Santosh Sanjeevannanavar, Nagamani A.N, "Efficient Design and FPGA Implementation of JPEG Encoder using Verilog HDL", IEEE, 2011.
2. Enas Dhuhri Kusuma, Thomas Sri Widodo "FPGA Implementation of Pipelined 2D-DCT and Quantization Architecture for JPEG Image Compression" IEEE, 2010.

3. L. Agostini, S. Bampi, "Pipelined Fast 2-D DCT Architecture for JPEG image Compression". IEEE Computer Society, 2001.
4. D. Trang, N. Bihn, "A High-Accuracy and High-Speed 2-D 8x8 Discrete Cosine Transform Design". Proceedings of ICGCRCICT 2010.
5. Gregory K. Wallace, "The JPEG Still Picture Compression Standard" IEEE, 1991.
6. Rafael C. Gonzalez, Richard E. Woods "Digital image processing" Prentice Hall publication, second edition.
7. Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, "Digital image processing using Matlab" Prentice Hall publication, First edition.
8. Thomas, Donald, Moorby, Phillip "The Veri log Hardware Description Language" Kluwer Academic Publishers, Norwell, MA.
9. Samir Palnitkar, "Veri log HDL, A Guide to Digital Design and Synthesis", Second Edition.
10. Xilinx, Inc., "Spartan-3 FPGA Family: Data Sheet", Xilinx Corporation.
11. Steve Kilts "Advanced FPGA Design" John Wiley & Sons, INC., Publications, First edition.
12. G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955.

